AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions of the claims and all prior listings of the claims in the present application.

1. (currently amended) A [[booth]] Booth processor, comprising:

a [[booth]] Booth recoder; and

a [[booth]] Booth register[[,]];

wherein an input to the [[booth]] Booth register is at least one output from the [[booth]] Booth recoder.

2. (currently amended) The [[booth]] Booth processor of claim 1, wherein the [[booth]] Booth register is a feedback register that stores at least one output value of the [[booth]] Booth recoder to be fed back to the ~~boothrecoder~~ Booth recoder.

3. (currently amended) The [[booth]] Booth processor of claim 2, wherein the at least one output value is a partial product selection signal, and

wherein the partial product selection signal is used to select a partial product value.

4. (currently amended) The [[booth]] Booth processor of claim 1, wherein the [[booth]] Booth register is a pipeline register, and

wherein the pipeline register stores output values of the [[booth]] Booth recoder.

5. (currently amended) A modulus processor, comprising:

a modulus recoder; and

a modulus feedback register[[,]];

wherein an input to the modulus feedback register is at least one output from the modulus recoder.

6. (original) The modulus processor of claim 5, wherein the modulus feedback register stores at least one output value of the modulus recoder to be fed back to the modulus recoder.

7. (currently amended) The modulus processor of claim 6 [[5]], wherein the at least one output value is a multiple modulus selection signal, and

wherein the multiple modulus selection signal is used to select a multiple modulus value.

8. (currently amended) A multiplier, comprising:

a [[booth]] Booth recoder;

a partial product synch register[[,]] ~~wherein an input to the partial product synch register is at least one output from the booth recoder~~;

a modulus recoder; and

a multiple modulus synch register[[,]];

wherein an input to the partial product synch register is at least one output from the Booth recoder,

wherein an input to the multiple modulus synch register is at least one output from the modulus recoder, and wherein the partial product synch register and the multiple modulus synch register are used to synchronize signals derived from the at least one output[[s]] of the [[booth]] Booth recoder and the at least one output of the modulus recoder.

9. (currently amended) The multiplier of claim 8, further comprising:

a [[booth]] Booth AND gate[[,]];

wherein at least one value from the partial product synch register is input to the [[booth]] Booth AND gate.

10. (currently amended) The multiplier of claim 8, further comprising:

a modulus AND gate[[,]];

wherein at least one value from the multiple modulus synch register is input to the modulus AND gate.

11. (currently amended) A multiplier, comprising:

a modulus recoder;

a modulus feedback register[[,]] ~~wherein an input to the modulus feedback register is at least one output from the modulus recoder~~;

a [[booth]] Booth recoder; and

a [[booth]] Booth register[[,]];

wherein an input to the modulus feedback register is at least one output from the modulus recoder, wherein an input to the [[booth]] Booth register is at least one output from the [[booth]] Booth recoder, and

wherein the modulus feedback register and the [[booth]] Booth register save values enabling decreased computation power usage in the multiplier.

12. (currently amended) The multiplier of claim 11, wherein the [[booth]] Booth register is a feedback register that stores at least one output value of the [[booth]] Booth recoder to be fed back to the [[booth]] Booth recoder.

13. (currently amended) The multiplier of claim 12, wherein the at least one output value is a partial product selection signal, and

wherein the partial product selection signal is used to select a partial product value.

14. (currently amended) The multiplier of claim 11, wherein the [[booth]] Booth register is a pipeline register, and

wherein the pipeline register stores output values of the [[booth]] Booth recoder.

15. (original) The multiplier of claim 11, wherein the modulus feedback register stores at least one output value of the modulus recoder to be fed back to the modulus recoder.

16. (currently amended) The multiplier of claim 15, wherein the at least one output value is a multiple modulus selection signal, and

wherein the multiple modulus selection signal is used to select a multiple modulus value.

17. (currently amended) The multiplier of claim 11, further comprising:

a [[booth]] Booth AND gate[[,]];

wherein at least one value from the [[booth]] Booth register is input to the [[booth]] Booth AND gate.

18. (currently amended) The multiplier of claim 11, further comprising:

a modulus AND gate[[,]];

wherein at least one value from the modulus feedback register is input to the modulus AND gate.

19. (currently amended) A partial product generator, comprising:

a [[booth]] Booth recoder; and

a mux[[,]];

wherein the mux inputs at least one output from the [[booth]] Booth recoder, and wherein the [[booth]] Booth recoder and the mux are used to obtain a partial product.

20. (currently amended) The partial product generator of claim 19, further comprising:

a [[booth]] Booth AND gate[[,]];

wherein at least one value from the mux is input to the [[booth]] Booth AND gate.

21. (currently amended) The partial product generator of claim 19, wherein the [[booth]] Booth recoder generates a partial product selection signal and a bit pattern is assigned to any

value of the partial product selection signal that is prohibited based on a previous value of the partial product selection signal.

22. (currently amended) The partial product generator of claim 21, wherein the bit pattern is chosen so that ~~the hamming~~ a Hamming distance between [[the]] a current value of the partial product selection signal and the previous value of the partial product selection signal is reduced.

23. (currently amended) The partial product generator of claim 21, wherein the bit pattern is chosen so that [[the]] an average temporal ~~hamming~~ Hamming distance between [[the]] current values of the partial product selection signal[[s]] and [[their]] corresponding previous values of the partial product selection signal are reduced.

24. (currently amended) The partial product generator of claim 21, wherein the [[booth]] Booth recoder ~~further~~ comprises:

a first mux[[,]]; and

a second mux;

wherein the first mux inputs a first portion of the previous value of the partial product selection signal and outputs a first portion of a current partial product selection signal[[;]], and

~~a second mux~~[[,]] wherein the second mux inputs a second portion of the previous value of the partial product selection signal and outputs a second portion of [[a]] the current partial product selection signal.

25. (original) The partial product generator of claim 24, wherein the first mux and the second mux are 8:1 muxs.

26. (currently amended) A multiple modulus generator, comprising:

a modulus recoder; and

a mux[[,]];

wherein if an enabling signal does not have a predetermined value, the modulus recoder generates a current multiple modulus selection signal ~~unless an enabling signal has a predetermined value~~,

wherein if the enabling signal [[has a]] does have the predetermined value, a previous value of [[the]] a multiple modulus selection signal is used without generating [[a]] the current multiple modulus selection signal, and

wherein the current multiple modulus selection signal or the previous value of the multiple modulus selection signal is used to select a multiple modulus value.

27. (currently amended) The multiple modulus generator of claim 26, further comprising:

a modulus AND gate[[,]];

wherein at least one value from the mux is input to the modulus AND gate.

28. (currently amended) The multiple modulus generator of claim 26, wherein the modulus recoder ~~further~~ comprises:

a first mux[[,]]; and

a second mux;

wherein the first mux inputs a first portion of the previous value of the multiple modulus selection signal and outputs a first portion of ~~acurrent~~ the current multiple modulus selection signal[[;]], and

~~a second mux~~[[,]] wherein the second mux inputs a second portion of the previous value of the multiple modulus selection signal and outputs a second portion of ~~acurrent~~ the current multiple modulus selection signal.

29. (original) The multiple modulus generator of claim 28, wherein the first mux and the second mux are 8:1 muxs.

30. (currently amended) A multiplier, comprising:

a modulus recoder;

a modulus feedback register[[,]] ~~wherein an input to the modulus feedback register is at least one output from the modulus recoder~~;

a modulus synch register[[,]] ~~wherein an input to the modulus synch register is at least one output from the modulus recoder~~;

a [[booth]] Booth recoder;

a [[booth]] Booth synch register[[,]] ~~wherein an input to the booth synch register is at least one output from the booth recoder~~; and

a [[booth]] Booth register[[,]] ~~wherein an input to the booth register is at least one output from the booth recoder~~[[,]];

wherein an input to the modulus feedback register is at least one first output from the modulus recoder,

wherein an input to the modulus synch register is at least one second output from the modulus recoder,

wherein an input to the Booth synch register is at least one first output from the Booth recoder,

wherein an input to the Booth register is at least one second output from the Booth recoder,

wherein the modulus feedback register and the [[booth]] Booth register save values enabling decreased computation power usage in the multiplier, and

wherein the [[booth]] Booth synch register and the modulus synch register are used to synchronize signals derived from the outputs of the [[booth]] Booth recoder and the modulus recoder to decrease glitches.

31. (currently amended) The multiplier of claim 30, wherein the [[booth]] Booth register is a feedback register that stores the at least one second output [[value]] of the [[booth]] Booth recoder to be fed back to the [[booth]] Booth recoder.

32. (currently amended) The multiplier of claim 31, wherein the at least one second output [[value]] is a partial product selection signal, and

wherein the partial product selection signal is used to select a partial product value.

33. (currently amended) The multiplier of claim 30, wherein the [[booth]] Booth register is a pipeline register, and

wherein the pipeline register stores output values of the [[booth]] Booth recoder.

34. (currently amended) The multiplier of claim 30, wherein the modulus feedback register stores the at least one first output [[value]] of the modulus recoder to be fed back to the modulus recoder.

35. (currently amended) The multiplier of claim 34, wherein the at least one first output [[value]] is a multiple modulus selection signal, and

wherein the multiple modulus selection signal is used to select a multiple modulus value.

36. (currently amended) The multiplier of claim 30, further comprising:

a [[booth]] Booth AND gate[[,]];

wherein at least one value from the [[booth]] Booth sync register is input to the [[booth]] Booth AND gate.

37. (currently amended) The multiplier of claim 30, further comprising:

a modulus AND gate[[,]];

wherein at least one value from the modulus ~~syncregister~~ sync register is input to the modulus AND gate.

38. (currently amended) The multiplier of claim 30, wherein a multiple modulus value and a partial product value are synchronized by using values from the modulus synch register and values from the [[booth]] Booth synch register.

39. (withdrawn—currently amended) A method of increasing computation speed of a radix 2^N Montgomery multiplication, where [[N>1]] N≥1, comprising:

providing inputs to a [[booth]] Booth recoder;

storing outputs of the [[booth]] Booth recoder; and

accumulating a result of the Montgomery multiplication[[,]];

wherein the storing and the accumulating are performed overlapped in time.

40. (withdrawn—currently amended) The method of claim 39, wherein the outputs of the [[booth]] Booth recoder are stored in a pipeline register.

41. (withdrawn) A method of reducing power consumption of a radix 2^N Montgomery multiplication, where N≥1, comprising:

receiving a modulus, multiplicator, and multiplicand;

synchronizing values related to the modulus, multiplicator, and multiplicand; and

accumulating the values to produce a result of the Montgomery multiplication.

42. (withdrawn—currently amended) The method of claim 41, further comprising:

calculating a multiple modulus using at least one of the modulus, multiplicator, and multiplicand; and calculating a partial product using at least one of the modulus, multiplicator, and multiplicand[[,]];

wherein the multiple modulus and the partial product are the synchronized values.

43. (withdrawn—currently amended) The method of claim 41, wherein [[the]] synchronizing ~~step further~~ values comprises:

storing at least two inputs related to the modulus, multiplicator, and multiplicand in synchronization registers.

44. (withdrawn—currently amended) The method of claim 42, further comprising:

matching [[the]] an arrival time of the multiple modulus and the partial product ~~and the multiple modulus~~ to an accumulator[[,]]; and

reducing overall power consumption of the Montgomery multiplication.

45. (withdrawn—currently amended) A method of reducing power consumption of a radix 2^N Montgomery multiplication, where [[$N>1$]] $N\geq1$, comprising:

providing inputs to a [[booth]] Booth recoder;

producing a selection signal using the [[booth]] Booth recoder;

assigning an inverted bit pattern to any value of the selection signal that is prohibited based on a previous value of the selection signal; and

storing outputs of the [[booth]] Booth recoder.

46. (withdrawn—currently amended) The method of claim 45, further comprising:

choosing the inverted bit pattern so that ~~the hamming~~ a Hamming distance between [[the]] a current value of the selection signal and the previous value of the selection signal is minimized.

47. (withdrawn—currently amended) The method of claim 45, wherein the inverted bit pattern is chosen so that [[the]] an average temporal ~~hamming~~ Hamming distance between [[the]] current values of the selection signal[[s]] and [[their]] corresponding previous values of the selection signal are minimized.

48. (withdrawn—currently amended) A method of reducing power consumption of a radix 2^N Montgomery multiplication, where $N\geq 1$, comprising:

determining an nth value of an iterative result signal;

providing an enable signal and the nth value of [[an]] the iterative result signal to a circuit;

if the enable signal renders [[the]] an (n+1)th value of the iterative result signal meaningless, then not calculating the (n+1)th value of the iterative result signal;

feeding back the nth value of the iterative result signal; and

using the nth value of the iterative result signal instead of the (n+1)th value of the iterative result signal.

49. (withdrawn—currently amended) The method of claim 48, wherein the nth and (n+1)th values of the iterative result signal [[is]] are determined by combinational logic.

50. (withdrawn—currently amended) The method of claim 49, wherein the combinational logic is performed by a [[MUX]] mux.

51. (withdrawn—currently amended) A method of reducing power consumption of a modulus ~~recorder~~ recoder, comprising:

determining an nth value of a multiple modulus selection signal;

storing the nth value of the multiple modulus selection signal in a register;

generating an (n+1)th enable signal[[,]] ~~wherein a predetermined value of the enable signal selects a multiple modulus value of zero~~; and

using the nth value of the multiple modulus selection signal without determining [[the]] an (n+1)th value of the multiple modulus selection signal if [[the]] a value of the (n+1)th enable signal is [[the]] a predetermined value;

wherein the predetermined value of the enable signal selects a multiple modulus value of zero.

52. (withdrawn—currently amended) The method of claim 51, further comprising:

selecting the multiple modulus value using the multiple modulus selection signal[[,]];

wherein ~~the step of~~ selecting the multiple modulus value is not performed when the value of the enable signal [[value]] is the predetermined value.

53. (currently amended) A Montgomery multiplier, comprising;

means for inputting, wherein the means for inputting[[,]] enters [[the]] values for a modulus, multiplicand, and a multiplier;

means for [[booth]] Booth storing, wherein the means for [[booth]] Booth storing stores at least one output value from a [[booth]] Booth recoder;

means for modulus storing, wherein the means for modulus storing stores at least one output value from a modulus recoder;

means for partial product generation, wherein the means for partial product generation produces a partial product value using [[the]] input from the means for input;

means for multiple modulus generation, wherein the means for multiple modulus generation produces a multiple modulus value using the input from the means for input;

means for synchronizing, wherein the means for synchronizing synchronizes the partial product value and [[the]] multiple modulus value; and

means for accumulating, wherein the means for accumulating inputs the synchronized partial product value and [[the]] multiple modulus value and produces a result for the Montgomery multiplier.